Wireless Access Point

- This application note is applicable to our **Ubuntu/Linux** Platforms.

### HostAP Mode Compatibility List

<table>
<thead>
<tr>
<th>Wifi module name (USB VID:PID)</th>
<th>ODROID-C1</th>
<th>ODROID-C2</th>
<th>ODROID-XU3/4</th>
<th>ODROID-N2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ubuntu 16.04</td>
<td>Ubuntu 18.04</td>
<td>Ubuntu 16.04</td>
<td>Ubuntu 16.04</td>
</tr>
<tr>
<td></td>
<td>v2.2-3.10.y</td>
<td>v3.0-3.10.y</td>
<td>v2.4-3.14.y</td>
<td>v3.0-3.16.y</td>
</tr>
<tr>
<td>0 - Ralink RT5370N (0x148F:0x5370)</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td></td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td></td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td></td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td>3 - Realtak RTL8188CUS (0x0BDA:0x8176)</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td></td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td></td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td></td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td>4 - Ralink RT5572N (0x148F:0x5572)</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td></td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td></td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td></td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td>5 - Realtak RTL8812AU (0x0BDA:0x8812)</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td></td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td></td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td></td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td>5A - Realtak RTL8811AU (0x0BDA:0xa811)</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td></td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td></td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td></td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
</tbody>
</table>

### wpa_supplicant Mode Compatibility List

<table>
<thead>
<tr>
<th>Wifi module name (USB VID:PID)</th>
<th>ODROID-C1</th>
<th>ODROID-C2</th>
<th>ODROID-XU3/4</th>
<th>ODROID-N2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ubuntu 16.04</td>
<td>Ubuntu 18.04</td>
<td>Ubuntu 16.04</td>
<td>Ubuntu 16.04</td>
</tr>
<tr>
<td></td>
<td>v2.2-3.10.y</td>
<td>v3.0-3.10.y</td>
<td>v2.4-3.14.y</td>
<td>v3.0-3.16.y</td>
</tr>
<tr>
<td>0 - Ralink RT5370N (0x148F:0x5370)</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td></td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td></td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
</tr>
<tr>
<td></td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
</tr>
<tr>
<td>3 - Realtak RTL8188CUS (0x0BDA:0x8176)</td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
</tr>
<tr>
<td></td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
</tr>
<tr>
<td></td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
</tr>
<tr>
<td></td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
</tr>
<tr>
<td>4 - Ralink RT5572N (0x148F:0x5572)</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td></td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td></td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td></td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
<td>confirm</td>
</tr>
<tr>
<td>5 - Realtak RTL8812AU (0x0BDA:0x8812)</td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
</tr>
<tr>
<td></td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
</tr>
<tr>
<td></td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
</tr>
<tr>
<td></td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
</tr>
<tr>
<td>5A - Realtak RTL8811AU (0x0BDA:0xa811)</td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
</tr>
<tr>
<td></td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
</tr>
<tr>
<td></td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
</tr>
<tr>
<td></td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
<td>not working</td>
</tr>
</tbody>
</table>

### ODROID Wireless Cards Information

Almost all the wireless card have capability of configuration as to be Access Point.

```bash
root@odroid:~# sudo iw list
```

If there is **“AP”** in the list of **“Supported interface modes”**, your device will support the Access
Point mode.

Supported interface modes:
* IBSS
* managed
* AP
* AP/VLAN
* WDS
* monitor
* mesh point

Refer to the below contents of informations of each wireless interface.

 Wifi Module 0

Manufacturer: Mediatek Ralink
Part name: RT5370N
Type: chip
Number of busses: 1
Bus(es): USB 2.0
Number of bands: 1
Band(s): 2.4GHz
Data rate: 150Mbps
MIMO configuration: 1x1:1 (1T1R)
IEEE 802.11 PHY Modes: b,g,n

Bus 005 Device 002: ID 148F:5370 Ralink Technology, RT5370 Wireless Adapter

 Wifi Module 3

Manufacturer: Realtek
Part name: RTL8188CUS
Type: chip
Number of busses: 1
Bus(es): USB 2.0
Number of bands: 1
Band(s): 2.4GHz
Data rate: 150Mbps
MIMO configuration: 1x1:1 (1T1R)
IEEE 802.11 PHY Modes: b,g,n

Bus 001 Device 003: ID 0bda:8176 Realtek Semiconductor Corp. RTL8188CUS 802.11n WLAN Adapter
Wifi Module 4

Manufacturer: Ralink
Part name: RT5572N
Type: chip
Number of busses: 1
Bus(es): USB 2.0
Number of bands: 2
Band(s): 2.4GHz, 5GHz
Data rate: 300Mbps
MIMO configuration: 2x2:2 (2T2R)
IEEE 802.11 PHY Modes: a,b,g,n

Bus 001 Device 006: ID 148f:5572 Ralink Technology, Corp. RT5572 Wireless Adapter

Wifi Module 5

Manufacturer: Realtek
Type: chip
Number of busses: 2
Bus(es): USB 2.0 / USB 3.0
Number of bands: 2
Band(s): 2.4GHz, 5GHz
Data rate: 300Mbps
MIMO configuration: 2x2:2 (2T2R)
IEEE 802.11 PHY Modes: a,b,g,n,ac

Bus 003: ID 0bda:8812 Realtek Semiconductor Corp. RTL8812AU 802.11a/b/g/n/ac WLAN Adapter

Wifi Module 5A

Manufacturer: Realtek
Type: chip
Number of busses: 1
Bus(es): USB 2.0
Number of bands: 2
Band(s): 2.4GHz, 5GHz
Data rate: AC600
MIMO configuration: 433 Mbps @ 5 GHz @ 1T1R / 150 Mbps @ 2.4 GHz @ 1T1R
IEEE 802.11 PHY Modes: a,b,g,n,ac

Bus 003 Device 003: ID 0bda:a811 Realtek Semiconductor Corp.
Configure Access Point

- One user (tam1111574) reported there's an issue with USB 3.0 port on the XU4:

- All commands must be executed in super user mode.
- You should do `apt update && apt full-upgrade` before proceed.
- It may need to do `apt install libnl-3-dev libnl-genl-3-dev libssl-dev hostapd iptables git pkg-config vim build-essential` to work with.

Configure wireless network interface and dnsmasq daemon.

Configure access point can be divided into the following tasks.

- Setup network interface configuration
- Setup DHCP server configuration
- Setup iptables to forward the internet traffic from Ethernet to wireless lan.
- Setup hostapd server or wpa_supplicant configuration.
- Check service & WIFI configuration

Step 1: Setup network interface configuration.

In order to configure Wireless Access Point you need to provide static IP address to Wireless network card.

Check `wlan0` part of the following contents that you should put.

```
root@odroid:~# vi /etc/network/interfaces
```

```
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp

auto wlan0
iface wlan0 inet static
    address 192.168.1.1
    netmask 255.255.255.0
```
**Note:** The predictable interface names shown as starts with `wlan` suppose you have an interface name.

If your wireless interface name shows like the below's,

```
root@odroid:~# ifconfig
enx7cdd9052131e Link encap:Ethernet  HWaddr 7c:dd:90:52:13:1e
  UP BROADCAST MULTICAST  MTU:1500  Metric:1
  RX packets: errors: dropped:201 overruns: frame:0
  TX packets: errors: dropped: overruns: carrier:0
  collisions: txqueuelen:1000
  RX bytes: (0.0 B)  TX bytes: (0.0 B)
```

Add `net.ifnames=0` as kernel parameter to kernel command line by editing the `boot.ini`. The local interface name issue gets resolved to have predictable name for wireless interface.

```
root@odroid:~# ifconfig
wlan0 Link encap:Ethernet  HWaddr 7c:dd:90:52:13:1e
  UP BROADCAST MULTICAST  MTU:1500  Metric:1
  RX packets: errors: dropped:10 overruns: frame:0
  TX packets: errors: dropped: overruns: carrier:0
  collisions: txqueuelen:1000
  RX bytes: (0.0 B)  TX bytes: (0.0 B)
```

**Note:** `wlan0` IP address might be different from yours.

**Note:** Wireless interface name can be changed as it depends on the wireless cards you installed.

### Step 2: Setup DHCP server configuration for Access Point.

Configure `dnsmasq` which is a light weight DHCP and DNS server.

```
root@odroid:~# apt install dnsmasq
Reading package lists... Done
Building dependency tree
Reading state information... Done
  upgraded, newly installed, 1 reinstalled, 0 removed, not upgraded.
Need to get 16.2 kB of archives.
After this operation, 0 B of additional disk space will be used.
Fetched 16.2 kB in 1s (15.6 kB/s)
(Reading database ... 155718 files and directories currently installed.)
Preparing to unpack .../dnsmasq_2.79-1_all.deb ...
Unpacking dnsmasq (2.79-1) over (2.79-1) ...
Setting up dnsmasq (2.79-1) ...
Processing triggers for systemd (237-3ubuntu10.9) ...
```
If you faced “FAILED” message when starting up the dnsmasq.service like the below “port 53: Address already in use”.

invoke-rc.d: initscript dnsmasq, action "start" failed.
- dnsmasq.service - dnsmasq - A lightweight DHCP and caching DNS server
  Loaded: loaded (/lib/systemd/system/dnsmasq.service; enabled; vendor preset: enabled)
  Active: failed (Result: exit-code) since Mon 2018-12-10 01:59:06 UTC; 22ms ago
  Process: 2073 ExecStart=/etc/init.d/dnsmasq systemd-exec (code=exited, status=2)
  Process: 2072 ExecStartPre=/usr/sbin/dnsmasq --test (code=exited, status=SUCCESS)

Dec 10 01:59:06 odroid systemd[1]: Starting dnsmasq - A lightweight DHCP and caching DNS server...
Dec 10 01:59:06 odroid dnsmasq[2073]: dnsmasq: syntax check OK.
Dec 10 01:59:06 odroid dnsmasq[2073]: dnsmasq: failed to create listening socket for port 53: Address already in use
Dec 10 01:59:06 odroid dnsmasq[2073]: failed to create listening socket for port 53: Address already in use
Dec 10 01:59:06 odroid dnsmasq[2073]: FAILED to start up
Dec 10 01:59:06 odroid systemd[1]: dnsmasq.service: Control process exited, code=exited status=2
Dec 10 01:59:06 odroid systemd[1]: dnsmasq.service: Failed with result 'exit-code'.
Dec 10 01:59:06 odroid systemd[1]: Failed to start dnsmasq - A lightweight DHCP and caching DNS server.
Processing triggers for systemd (237-3ubuntu10.9) ...

Stop the service listening port 53.

# Check that systemd-resolve service is listening port 53 now(127.0.0.53:53)
root@odroid:~# netstat -alnp | grep -w LISTEN
tcp 127.0.0.53:53 LISTEN 0.0.0.0:*
LISTEN 755/systemd-resolve tcp 0.0.0.0:22 LISTEN 0.0.0.0:*
LISTEN 916/sshd tcp 127.0.0.1:531 LISTEN 0.0.0.0:*
LISTEN 2616/cupsd tcp6 :::22 LISTEN :::
LISTEN 916/sshd tcp6 :::1:531 LISTEN :::
LISTEN 2616/cupsd

# To use the 53 port, disable & stop the systemd-resolved service
root@odroid:~# systemctl disable systemd-resolved.service
Removed /etc/systemd/system/multi-user.target.wants/systemd-
resolved.service.
Removed /etc/systemd/system/dbus-org.freedesktop.resolve1.service.

root@odroid:~# systemctl stop systemd-resolved

# dnsmasq service enable & start
root@odroid:~# systemctl enable dnsmasq
Synchronizing state of dnsmasq.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable dnsmasq

root@odroid:~# systemctl start dnsmasq
# confirm dnsmasq service running
root@odroid:~# netstat -alnp | grep -w LISTEN
tcp 0.0.0.0:53 0.0.0.0:*
LISTEN 6100/dnsmasq
tcp 0.0.0.0:22 0.0.0.0:*
LISTEN 677/sshd
tcp 127.0.0.1:631 0.0.0.0:*
LISTEN 2734/cupsd
tcp6 :::53 ::::*
LISTEN 6100/dnsmasq
tcp6 :::22 ::::*
LISTEN 677/sshd
tcp6 :::1.0.0.1 ::::*
LISTEN 2734/cupsd

Copy **dnsmasq** configuration file to get a backup and then make a new one.

root@odroid:~# mv /etc/dnsmasq.conf /etc/dnsmasq.conf.org
root@odroid:~# vi /etc/dnsmasq.conf

dnsmasq.conf

domain-needed
bogus-priv
no-resolv
no-poll
server=/example.com/192.168.1.5
server=8.8.8.8
server=8.8.4.4
local=/example.com/
address=/doubleclick.net/127.0.0.1
no-hosts
#adn-hosts=/etc/dnmasq.d/hosts.conf
expand-hosts
domain=example.com
dhcp-range=192.168.1.20,192.168.1.50,72h
dhcp-range=tftp,192.168.1.250,192.168.1.254
```
dhcp-option=option:router,192.168.1.1  
dhcp-option=option:ntp-server,192.168.1.5  
dhcp-option=19, # ip-forwarding off  
dhcp-option=44,192.168.1.5 # set netbios-over-TCP/IP aka WINS  
dhcp-option=45,192.168.1.5 # netbios datagram distribution server  
dhcp-option=46,8 # netbios node type
```

**Note:** You can extend the dhcp-range or change the IP address in the configuration.

### Step 3: Setup iptables to forward the internet traffic from Ethernet to wireless LAN

Next, make port forwarding enabled automatically on boot up.

```
root@odroid:~# vi /etc/sysctl.conf
```

Find the options below and change as them.

```
net.ipv4.ip_forward=1  
network.ipv6.conf.all.forwarding=1
```

Add the following contents to /etc/rc.local file in order to redirect internet traffic to wireless lan.

```
root@odroid:~# vi /etc/rc.local
```

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
```

**Note:** Update the iptables rules depending on your wireless network interface.

Reboot the system to get work.

```
root@odroid:~# reboot
```

### Step 4: Access Point Installation - hostapd

#### Step 4-1: Setup Hostapd server configuration

hostapd is a user space daemon for access point and authentication servers. It implements IEEE 802.11 access point management, IEEE 802.1X/WPA/WPA2/EAP Authenticators, RADIUS client, EAP server, and RADIUS authentication server.

Steps to get the hostapd compilation. Download the hostapd source code and build with proper parameters.

**Wifi Module 0, 4, 5, 5A**

```
root@odroid:~# wget https://w1.fi/releases/hostapd-2.6.tar.gz
root@odroid:~# tar xvfz hostapd-2.6.tar.gz
root@odroid:~# cd hostapd-2.6/
root@odroid:~/hostapd-2.6/hostapd# cp defconfig .config
root@odroid:~/hostapd-2.6/hostapd# echo CONFIG_LIBNL32=y >> .config
root@odroid:~/hostapd-2.6/hostapd# echo CONFIG_IEEE80211N=y >> .config
root@odroid:~/hostapd-2.6/hostapd# echo CONFIG_IEEE80211AC=y >> .config
root@odroid:~/hostapd-2.6/hostapd# make
```

**Wifi Module 3**

You should download a patch file if you use Wifi Module 3 to make hostapd works for Wifi Module 3 which do not support the standard nl80211 driver from hostapd on kernel 3.10, 3.14, 3.16.

```
root@odroid:~# git clone https://github.com/pritambaral/hostapd-rtl871xdrv.git
root@odroid:~# wget https://w1.fi/releases/hostapd-2.6.tar.gz
root@odroid:~# tar xvfz hostapd-2.6.tar.gz
root@odroid:~# cd hostapd-2.6
root@odroid:~/hostapd-2.6# patch -p1 < ../hostapd-rtl871xdrv/rtlxdrv.patch
root@odroid:~/hostapd-2.6# cd hostapd
root@odroid:~/hostapd-2.6/hostapd# cp defconfig .config
root@odroid:~/hostapd-2.6/hostapd# echo CONFIG_LIBNL32=y >> .config
root@odroid:~/hostapd-2.6/hostapd# echo CONFIG_DRIVER_RTW=y >> .config
root@odroid:~/hostapd-2.6/hostapd# make
```

Backup the current hostapd. Replace the file with the configured one.

```
root@odroid:~/hostapd-2.6/hostapd# mv /usr/sbin/hostapd /usr/sbin/hostapd.org
root@odroid:~/hostapd-2.6/hostapd# cp hostapd /usr/sbin/
```

Verify that you have installed the latest version.

```
root@odroid:~/hostapd-2.6/hostapd# cd ~
root@odroid:~# hostapd
hostapd v2.6
User space daemon for IEEE 802.11 AP management,
```
Configure hostapd for Access Point use

```
root@odroid:~# vi /etc/default/hostapd
```

Find the options below and edit.

```plaintext
DAEMON_CONF="/etc/hostapd/hostapd.conf"
DAEMON_OPTS="-B"
```

**Note:** You can update this DAEMON_OPTS to get the logs from hostapd daemon.

Edit `/etc/hostapd/hostapd.conf` file.

```
root@odroid:~# vi /etc/hostapd/hostapd.conf
```

---

- If you have trouble with this settings with Wifi module 3, check out the below list.
  - You should build your `rtl871xdrv patched hostapd` and copy that results to `/usr/sbin` directory.
  - Don't forget to comment the codes out in N-WLAN categories.
  - `rtl8192cu, rtl_usb, rtl8192c_common, rtlwifi` modules should be loaded.
  - Or if it still doesn't work and/or if you use kernel version under 4.4, you can try to follow the previous guide: https://wiki.odroid.com/accessory/connectivity/wifi/wlan_ap
  - Also you can try to use a pre-built script: https://github.com/oblique/create_ap

---

Hostapd configuration for __2.4 Ghz__ configuration

```
hostapd.conf

# HostAPD <2.4 Ghz configuration hostapd.conf file>
# Interface
interface=wlan0
# driver
driver=nl80211
# Logging
logger_syslog=-1
logger_syslog_level=3
logger_stdout=-1
logger_stdout_level=2
# CTRL-Interface
```

http://wiki.odroid.com/
**Hostapd configuration for __5 Ghz__ Realtek RTL8812AU chipset**

**hostapd.conf**

```plaintext
ctrl_interface=/var/run/hostapd
ctrl_interface_group=

# WLAN
country_code=KR
ssid=OdroidAPn
ew_mode=g
channel=6
beacon_int=100
dtim_period=2
max_num_sta=255
rts_threshold=2347
fragm_threshold=2346
preamble=1

# WPA2
wpa=2
wpa_passphrase=hardkernel
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
auth_algs=3
macaddr_acl=
wmm_enabled=1
eap_reauth_period=360000
fragm_threshold=2346
rsn_preauth=1
rsn_preauth_interfaces=wlan0
wpa_group_rekey=600
wpa_ptk_rekey=600
wpa_gmk_rekey=86400

##### If you use Wifi module 3 (RTL8188CUS), you have to comment all the below contents out #####

# N-WLAN
ieee80211n=1
ht_capab=[HT20+][SHORT-GI-20][DSSS_CCK-20]
require_ht=
obss_interval=
```

---

ssid=Odroid5Ghz
hw_mode=a
channel=36
max_num_sta=128
auth_algs=1
### DFS
country_code=KR
ieee80211d=1
ieee80211h=1
#ieee80211lac=1
### IEEE 802.11n
ieee80211n=1
ht_capab=[HT40+] [SHORT-GI-20] [SHORT-GI-40] [DSSS_CCK-20]
### IEEE 802.11ac
#ieee80211lac=1
#vht_oper_chwidth=1
#vht_capab=[SHORT-GI-40] [RXLDPC] [TX-STBC-2BY1]
#vht_oper_centr_freq_seg0_idx=0
### IEEE 802.11i
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_passphrase=hardkernel
rsn_pairwise=CCMP
### hostapd event logger
logger_syslog=-1
logger_syslog_level=2
logger_stdout=-1
logger_stdout_level=2
### WMM
wmm_enabled=1
uapsd_advertisement_enabled=1
wmm_ac_bk_cwmin=4
wmm_ac_bk_cwmax=10
wmm_ac_bk_aifs=7
wmm_ac_bk_txop_limit=
wmm_ac_be_aifs=3
wmm_ac_be_cwmin=4
wmm_ac_be_cwmax=10
wmm_ac_be_txop_limit=
wmm_ac_be_acm=
wmm_ac_vi_aifs=2
wmm_ac_vi_cwmin=3
wmm_ac_vi_cwmax=4
wmm_ac_vi_txop_limit=94
wmm_ac_vi_acm=
wmm_ac_vo_aifs=2
wmm_ac_vo_cwmin=2
wmm_ac_vo_cwmax=3
wmm_ac_vo_txop_limit=47
### TX queue parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>tx_queue_data3_aifs</td>
<td>7</td>
</tr>
<tr>
<td>tx_queue_data3_cwmin</td>
<td>15</td>
</tr>
<tr>
<td>tx_queue_data3_cwmax</td>
<td>1023</td>
</tr>
<tr>
<td>tx_queue_data3_burst</td>
<td></td>
</tr>
<tr>
<td>tx_queue_data2_aifs</td>
<td>3</td>
</tr>
<tr>
<td>tx_queue_data2_cwmin</td>
<td>15</td>
</tr>
<tr>
<td>tx_queue_data2_cwmax</td>
<td>63</td>
</tr>
<tr>
<td>tx_queue_data2_burst</td>
<td></td>
</tr>
<tr>
<td>tx_queue_data1_aifs</td>
<td>1</td>
</tr>
<tr>
<td>tx_queue_data1_cwmin</td>
<td>7</td>
</tr>
<tr>
<td>tx_queue_data1_cwmax</td>
<td>15</td>
</tr>
<tr>
<td>tx_queue_data1_burst</td>
<td>3.0</td>
</tr>
<tr>
<td>tx_queue_data0_aifs</td>
<td>1</td>
</tr>
<tr>
<td>tx_queue_data0_cwmin</td>
<td>3</td>
</tr>
<tr>
<td>tx_queue_data0_cwmax</td>
<td>7</td>
</tr>
<tr>
<td>tx_queue_data0_burst</td>
<td>1.5</td>
</tr>
</tbody>
</table>

**Note:** On the Wifi module 4 and Wifi module 5, Wifi module 5A, we need to tune these setting depending on the ht_capab.

**Note:** Replace the country_code, ssid, and wpa_passphrase as per your requirement.

**Note:** N-WLAN is optional that can be dropped out.

### Step 4-2: Check your hostpad & dnsmasq service status

- WIFI IP address must be the same of the set for `/etc/network/interfaces`.

You have to allocate specific IP address you set into the `/etc/network/interfaces` file as a static method to wlan0 interface.

```bash
# Force allocation of IP address
root@odroid:~# ifconfig wlan0 192.168.1.1
```

```bash
# Check IP address
root@odroid:~# ifconfig wlan0
wlan0: flags=883<UP, BROADCAST, RUNNING, MULTICAST> mtu 1500
       inet 192.168.1.1 netmask 255.255.255.0 broadcast 192.168.1.255
       ether 40:a5:ef:f3:98:6a txqueuelen 1000 (Ethernet)
       RX packets 122 bytes 13344 (13.3 KB)
       RX errors 0 dropped 0 overruns 0 frame 0
       TX packets 49 bytes 18722 (18.7 KB)
```
After you finish all of the setup steps, restart the services and check their statuses.

Enter the following code to enable by default.

```
root@odroid:~# update-rc.d hostapd enable
```

Restart the services.

```
root@odroid:~# service hostapd restart
root@odroid:~# service dnsmasq restart
```

- If you get this message when starting hostapd service, “Failed to start hostapd.service: Unit hostapd.service is masked.”, unmask that service and try it out again.

```
systemctl unmask hostapd.service
```

Check the services.

```
root@odroid:~# service hostapd status
root@odroid:~# service dnsmasq status
```

Check **active (running)** status as shown as the followings.

### hostapd
- hostapd.service - LSB: Advanced IEEE 802.11 management daemon
  - Loaded: loaded (/etc/init.d/hostapd; bad; vendor preset: enabled)
  - Active: active (running) since Sun 2017-07-09 19:48:41 UTC; 31s ago
    - Docs: man:systemd-sysv-generator(8)
    - Process: 789 ExecStart=/etc/init.d/hostapd start (code=exited, status=/SUCCESS)
    - CGroup: /system.slice/hostapd.service
      - /usr/sbin/hostapd -B -P /run/hostapd.pid -B /etc/hostapd/hostapd-2.6.conf

Jul 09 19:48:41 odroid systemd[1]: Starting LSB: Advanced IEEE 802.11 management daemon...
Jul 09 19:48:41 odroid hostapd[789]: * Starting advanced IEEE 802.11 management hostapd
Jul 09 19:48:41 odroid hostapd[789]: ...done.

### dnsmasq
- dnsmasq.service - dnsmasq - A lightweight DHCP and caching DNS server
Step 4: Access Point Installation - wpa_supplicant

Step 4-1: Setup wpa_supplicant configuration

wpa_supplicant is a user space application which works as a WPA supplicant and SME (to handle initiating MLME commands).

Please refer to the link for further informations:

Steps to get the wpa_supplicant compilation. Download the wpa_supplicant source code and build.

```
root@odroid:~# wget https://w1.fi/releases/wpa_supplicant-2.6.tar.gz
root@odroid:~# tar xvfz wpa_supplicant-2.6.tar.gz
```
Backup the current wpa_supplicant. Replace the file with configured one.

Verify that you have installed the latest version.

Create a config file for wpa_supplicant to run in AP mode. We're going to create that names wpa.conf.

Reboot the system.

Note: You can change the frequency range between 2.4 GHz and 5 GHz.
Confirm them.

```
root@odroid:~# ifconfig wlan0
wlan0:   flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
          inet 192.168.1.1  netmask 255.255.255.0  broadcast 192.168.1.255
          ether 40:a5:ef:f3:98:6a  txqueuelen 1000 (Ethernet)
          RX packets 122  bytes 13344  (13.3 KB)
          RX errors  dropped  overruns  frame
          TX packets 49  bytes 18722  (18.7 KB)
          TX errors  dropped  overruns  carrier  collisions
```

This is an example of starting wpa_supplicant as Access Point.

```
root@odroid:~# wpa_supplicant -Dnl80211 -iwlan0 -c ~/wpa.conf
Successfully initialized wpa_supplicant
Using interface wlan0 with hwaddr 7c:dd:90:52:13:1e and ssid "odroid-WPA"
```

```
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
wlan0: CTRL-EVENT-CONNECTED - Connection to 7c:dd:90:52:13:1e completed [id=
        id_str=
```

**Start Wpa_supplicant as a server**

Append the following in /etc/rc.local file

```
sudo service network-manager stop
wpa_supplicant -B -Dnl80211 -iwlan0 -c /root/wap.conf
```

**Alternative/Easier Ways For Access Point Configuration**

This script creates a NATed or Bridged WiFi Access Point.

From:  
http://wiki.odroid.com/ - ODROID Wiki

Permanent link:  
http://wiki.odroid.com/accessory/connectivity/wifi/wireless_ap_mode

Last update: 2019/03/05 01:05